

Plataforma de serviços de dados abertos do TCU: *crowdsourcing*, nuvem cívica e aplicativos cívicos



Monique Louise de Barros Monteiro

é servidora do Tribunal de Contas da União, mestre e bacharel em Ciência da Computação pela UFPE. Possui 12 certificações profissionais: Big Data Certified Science Professional, Big Data Certified Professional, SOA Certified Governance Specialist, SOA Certified Architect, SOACP, PSM, SAFe Agilist, Oracle Master Java EE 5 Enterprise Architect, MTS, MSP, SCJP e IBM OOAD.



Marcelo Pacote

é servidor do Tribunal de Contas da União, mestre e bacharel em Ciência da Computação pela UnB. Possui especialização em Desenvolvimento de Aplicativos Móveis pela Uniara, além de 17 certificações profissionais: PMP, CBDP, SOACP, SOACC, CSM, CSD, SCJD, SCJP, SCJA, SCEA (I), SCBCD, SCWCD, RUPF, IRIP, CTFI, ITILF e Oracle SQL Expert. Autor de livro na área de Tecnologia da Informação.



Remis Balaniuk

é servidor do Tribunal de Contas da União, pós-doutor em Realidade Virtual pela Stanford University, doutor em Informática pelo Institut National Polytechnique de Grenoble - França, mestre em Ciências da Computação pela Universidade Federal do Rio Grande do Sul e tecnólogo em Processamento de Dados pela Universidade de Brasília. Atualmente é professor e pesquisador da Universidade Católica de Brasília.



RESUMO

Este artigo descreve o trabalho de desenvolvimento, disponibilização e sustentação de uma plataforma de exposição de dados abertos e de coleta de informações por meio de *crowdsourcing*, tendo por base seu uso por aplicativos cívicos. Como resultados esperados da implementação dessa plataforma, podemos citar as oportunidades de 1) centralização e disponibilização de dados abertos e 2) obtenção de contribuições e percepções dos cidadãos. Tais informações poderão ter grande utilidade tanto para a definição de indicadores para serviços e políticas públicas como também para aumentar o conhecimento a respeito dos anseios do cidadão em diferentes localidades e esferas populacionais, incluindo a percepção da sociedade com relação aos serviços a ela prestados.

Palavras-chave: aplicativos cívicos, *web services*, *crowdsourcing*, nuvem cívica, dados abertos

1. INTRODUÇÃO

Um tema recorrente na Administração Pública no Brasil e em outras nações é o compartilhamento de dados abertos para maior transparência em relação às ações governamentais.

Atualmente, temos no Brasil, por exemplo, o Portal Brasileiro de Dados Abertos, ferramenta disponibi-

lizada pelo governo para que todos possam encontrar e utilizar os dados e as informações públicas. O portal inclui dados relativos à execução orçamentária, às informações cartográficas e sobre unidades de atendimento, além de possuir indicadores e estatísticas diversas.

Dentre os fatores responsáveis por catalisar iniciativas como o **Portal Brasileiro de Dados Abertos**¹ está a Lei 12.527 – Lei de Acesso à Informação Pública, sancionada em 18 de novembro de 2011. De acordo com as diretrizes fixadas pelo dispositivo legislativo em questão, a regra geral consiste em classificar as informações de interesse coletivo produzidas ou custodiadas pelo Estado como públicas, visando a assegurar o direito fundamental de acesso à informação. Assim, informações devem ser classificadas como sigilosas em casos excepcionais, quando for imprescindível à segurança da sociedade e do Estado.

O Tribunal de Contas da União (TCU) tem em seu planejamento ações para promover a divulgação de dados abertos por meio dos seus veículos de informação. No contexto desse conjunto de ações, desde o início de 2015, o TCU tem internalizado bases de dados provenientes do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), do Cadastro Nacional de Estabelecimentos de Saúde (CNES)², do Ministério do Desenvolvimento Social e Agrário (relativas à assistência social), do Sistema Nacional de Emprego (SINE), do Portal de Compras do Governo Federal (SIASG/Comprasnet)³, do Sistema Integrado de Admi-

nistração Financeira do Governo Federal (SIAFI)⁴, do Cadastro Nacional de Empresas (CNE)⁵, dentre outros.

Após a internalização das bases, os dados recebidos são disponibilizados para a sociedade, por meio de consulta por usuários finais ou por sistemas de software. A seguir, é detalhada a metodologia adotada para esta disponibilização dos chamados dados abertos.

2. METODOLOGIA

Tradicionalmente, ferramentas como o Portal Brasileiro de Dados Abertos, apesar de sua utilidade, têm apresentado como característica principal a disponibilização das informações de forma estática e/ou bruta (ex.: planilhas, arquivos em formato XML, CSV ou HTML), com grande quantidade de registros (ex.: planilhas com dezenas de milhares de linhas ou arquivos com centenas de *megabytes*). Porém, dados eletrônicos governamentais brutos são, em geral, de difícil compreensão para o cidadão comum. Faz-se necessária, portanto, a intervenção de pessoas, grupos e/ou empresas, com capacidade e disponibilidade para desenvolver aplicativos e outros elementos tecnológicos que possam traduzir o dado governamental aberto em produtos de interesse e utilidade pública ou privada.

Grande parte dos conjuntos de dados disponibilizados é de difícil apropriação também pelos desenvolvedores de tecnologias cívicas, já que essas bases de dados, oferecidas tipicamente em formato bruto, comumente têm um baixo valor informativo, além de serem disponibilizadas na forma de arquivos estáticos. Tal característica exige do desenvolvedor um trabalho periódico de transferência, entendimento, tratamento, adequação e posterior disponibilização em aplicativos. Também é requisito comum a necessidade de infraestrutura de TI remota e disponível continuamente, o que implica em custos de hospedagem (ex.: contratação de serviços de nuvens) com os quais os desenvolvedores precisam arcar. Tais dificuldades limitam o potencial de uso de dados abertos pelos aplicativos cívicos.

Ciente desse cenário, o TCU adotou um modelo de atuação no ecossistema de dados abertos que assumiu como premissa a necessidade de oferecer aos desenvolvedores de tecnologias cívicas uma plataforma de serviços na qual seus aplicativos possam acessar remotamente dados tratados e atualizados de uma forma mais adequada e otimizada para processamento por sistemas de software clientes. Outra premissa é que a plataforma oferecida permita aos aplicativos **armazenar** os dados por eles gerados.

Primeiramente, os dados recebidos a partir das bases citadas na introdução são persistidos em um banco relacional corporativo.

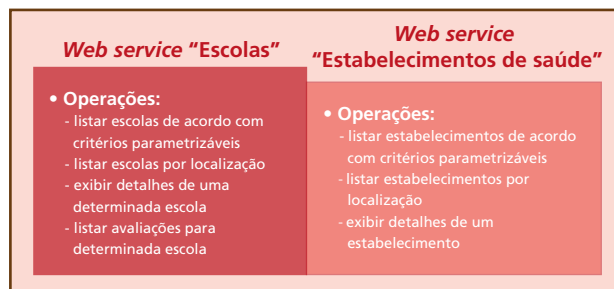
Concluída a etapa de persistência dos dados na infraestrutura de armazenamento do TCU, eles estão prontos para serem acessados tanto por usuários finais como por aplicativos através de *web services*. Segundo Erl et al. (2012), “a partir de uma perspectiva geral, um *serviço* é um programa de software que disponibiliza sua funcionalidade por meio de uma interface técnica publicada, denominada *contrato do serviço*”. Considerando que tais serviços de software podem ser implementados por meios diversos, utilizaremos neste artigo a denominação *web service* para nos referirmos a serviços disponíveis na *World Wide Web*.

Cada *web service* é responsável por agrupar e expor um conjunto de operações relacionadas. Tais operações podem ser acessadas por aplicações desenvolvidas externamente para a obtenção dos dados por meio de protocolos padronizados, de acordo com os padrões da *World Wide Web*. Tais protocolos incluem, por exemplo, *Hypertext Transfer Protocol* (HTTP) (FIELDING et al., 1999).

A figura 1 mostra o conceito de *web service* e suas operações em nível conceitual. Temos dois *web services* representados e cada um deles responde por determinado “tema”, agrupando operações de consulta que serão executadas sobre a base de dados persistida. Cada operação, por sua vez, admite o recebimento de parâmetros que permitem filtrar, dinamicamente, os crité-

Figura 1:

Representação em alto nível de dois *web services*. Um agrupa operações de consulta sobre escolas, e outro, operações de consulta sobre estabelecimentos de saúde





rios de busca, bem como os atributos e a quantidade de dados retornados.

Com relação ao formato dos dados retornados pelas operações, utilizamos JSON – *JavaScript Object Notation* (ECMA, 2013). Tal formato é compreensível por usuários finais, uma vez que é textual. Além disso, por ser semiestruturado, é amplamente utilizado por aplicativos para dispositivos móveis e outros tipos de sistemas computacionais, garantindo ampla interoperabilidade e facilidade de manipulação por desenvolvedores, principalmente em comparação com formatos proprietários, binários ou de maior complexidade, comumente encontrados em portais tradicionais de dados abertos.

3. DESENVOLVIMENTO

Os *web services* foram projetados inicialmente como ferramenta de apoio para o desenvolvimento de aplicativos móveis cívicos, no contexto do Projeto BE-PiD (Programa Educacional Brasileiro de Desenvolvimento para IOS), por meio de uma parceria entre o TCU e a Universidade Católica de Brasília (UCB)⁶. A parceria em questão envolveu o desenvolvimento de aplicativos cívicos por parte de alunos da referida universidade, que utilizassem os canais de informação fornecidos pelos *web services* desenvolvidos pelo TCU.

Foram especificados e desenvolvidos conjuntos de *web services* para disponibilização dos seguintes grupos de informações:

- **Saúde:** informações de especialidades médicas, estabelecimentos de saúde, profissionais, medicamentos e serviços especializados, todos no âmbito de serviços públicos.
- **Assistência social:** informações de centros de referência e centros de referência especializada de assistência social (respectivamente, CRAS e CREAS).
- **Empregos:** informações de postos do SINE.
- **Educação:** informações de educação, incluindo basicamente dados cadastrais dos estabelecimentos educacionais e de seus recursos.

Denominaremos o conjunto de *web services* acima citados de “nuvem cívica”. Daqui em diante, para nos referirmos especificamente ao conjunto de operações definidas no contrato ou interface disponibilizado pelos *web services*, utilizaremos o acrônimo API, sigla para *Application Programming Interface*.

A API inicialmente proposta e disponibilizada seguiu os requisitos esperados pelos aplicativos “Mapa da Saúde” e “Nossa Escola”, desenvolvidos por alunos da UCB.

Em virtude principalmente das necessidades de coletar *feedback* de cidadãos – prática conhecida mundialmente como *crowdsourcing* – os aplicativos foram desenvolvidos seguindo um modelo de rede social, de-

mandando, portanto, a existência de um cadastro de usuários e de avaliações de serviços públicos por eles fornecidas. A partir daí, surgiu o **metamodelo para aplicativos cívicos**, um conjunto adicional de *web services* que permite consultar, incluir, alterar e excluir informações relacionadas a usuários, aplicativos, grupos de usuários, *postagens*, *hashtags*, notificações etc.

Adicionalmente, para que o mesmo modelo pudesse ser adotado tanto para saúde pública como para educação, bem como para outras áreas de atuação do Estado, é possível que usuários enviem avaliações, sob a forma de **postagens**, de entidades genéricas denominadas aqui de **objetos**. Assim, um usuário pode avaliar um objeto, que pode ser, por exemplo, uma escola ou um hospital, a depender do contexto de utilização determinado pelo aplicativo em uso.

As massas de dados geradas pelo *crowdsourcing* cívico serão fontes inestimáveis de informação, em particular para a concepção de indicadores que permitirão a descoberta de conhecimento a respeito do funciona-

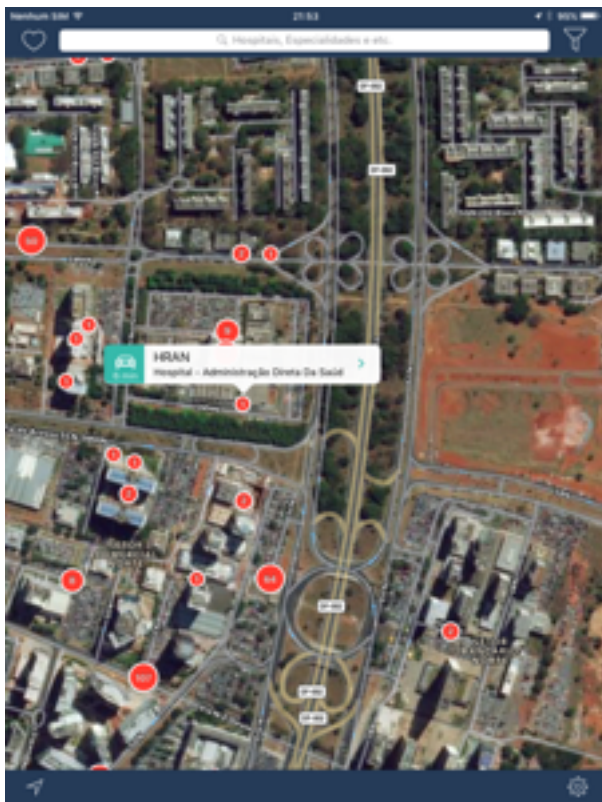
mento da máquina administrativa pública, das necessidades e da percepção do cidadão. O *crowdsourcing*, quando utilizado como uma prática de transparência, permite que grandes quantidades de dados ganhem sentido, podendo gerar ideias novas para o desenvolvimento de projetos para sociedade.

Esse conhecimento servirá futuramente como insumo no planejamento de ações de controle do TCU. Por fim, a coleta e o armazenamento dos dados em um formato definido pelo próprio TCU facilita iniciativas futuras de análise estatística e mineração de dados, tanto preditiva como prescritiva, podendo incluir desde análises mais simples até aplicações mais complexas baseadas em aprendizado de máquina.

Como exemplo de aplicativo cívico, as figuras 2 a 3 mostram capturas de tela do aplicativo Mapa da Saúde, que possui recursos de busca por estabelecimentos de saúde em um raio de localização, exibição de dados detalhados de cada estabelecimento (incluindo, embora não seja exibido nas capturas, listagem de especialida-

Figura 2:

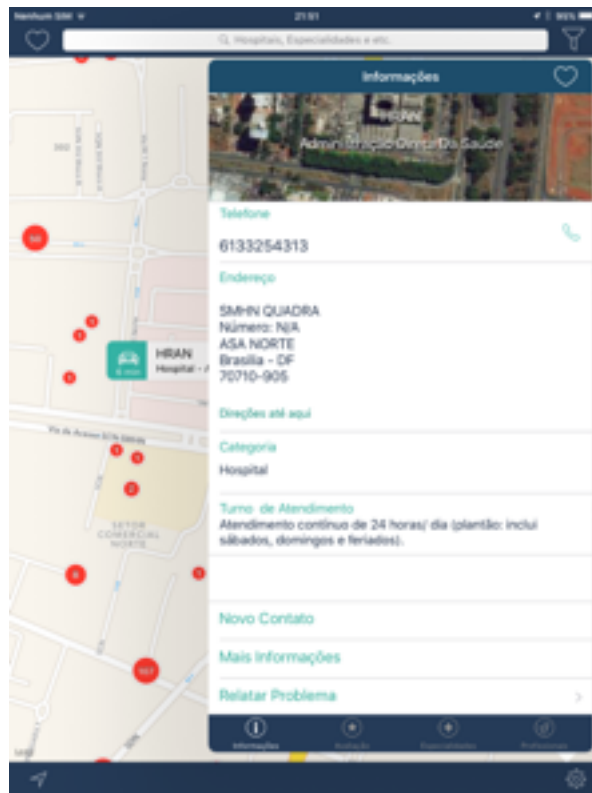
Mapa da Saúde – busca de estabelecimentos de saúde em raio, exibidos em um mapa (visualização em modo satélite)



Fonte: Elaboração própria

Figura 3:

Mapa da Saúde – detalhe de um estabelecimento



Fonte: Elaboração própria

des médicas e quantitativo de profissionais por função) e ainda uma funcionalidade para que o usuário avalie a qualidade do atendimento prestado.

4. ASPECTOS TECNOLÓGICOS

A nuvem cívica foi desenvolvida segundo o estilo REST – sigla para *Representational State Transfer* (FIELDING, 2000). REST engloba um estilo arquitetônico para a construção de aplicações e APIs e, grosso modo, provê um mecanismo no qual, no contexto dos *web services* aqui referenciados, cada operação de um *web service* pode ser representada da seguinte forma:

Método HTTP (GET | POST | PUT | DELETE) + URI (*Uniform Resource Identifier*)

Cada um dos **métodos** acima representa um dos possíveis métodos ou operações suportados pelo protocolo HTTP, a serem executados sobre um **recurso** identificado por uma **URI**. Os métodos possuem a seguinte semântica, resumidamente:

- GET: Utilizado para recuperar informações (ex.: operações de consulta);
- POST: Utilizado para criar novos recursos (ex.: cadastrar novo usuário);
- PUT: Utilizado para atualizar recursos (ex.: atualizar dados cadastrais de um usuário já existente);
- DELETE: Utilizado para excluir recursos (ex.: excluir um usuário).

Exemplos de chamadas a operações da API que podem ser executadas por aplicativos:

- GET <http://contas.tcu.gov.br/nossaEscolaRS/escolas/123>: recupera as informações da escola identificada externamente pela URL <http://contas.tcu.gov.br/nossaEscolaRS/escolas/123>. Considera-se neste exemplo que “123” é o identificador da escola interno à aplicação (ex.: chave em um banco de dados);
- DELETE <http://contas.tcu.gov.br/appCivicoRS/pessoas/456>: excluir o usuário identificado externamente pela URI <http://contas.tcu.gov.br/appCivicoRS/pessoas/456>. Considera-se neste

exemplo que “456” é o identificador da escola interno à aplicação (ex.: chave em um banco de dados);

- POST <http://contas.tcu.gov.br/appCivicoRS/pessoas> {“nome”: “José da Silva”, “email”: “josedasilva@tcu.gov.br”, “dataNascimento”: “12/12/1950”}: cadastra um novo usuário, enviando suas informações no corpo da mensagem;
- PUT <http://contas.tcu.gov.br/appCivicoRS/pessoas/789> {“nome”: “José da Silva”, “email”: “novoemail@tcu.gov.br”, “dataNascimento”: “12/12/1950”}: atualiza os dados de um usuário existente, enviando suas informações no corpo da mensagem.

A API é documentada no formato Swagger⁷. Tal formato permite não apenas a visualização do contrato da nuvem cívica (operações suportadas, formatos das URIs, formato dos dados trafegados, parâmetros para cada operação) como também o teste das funcionalidades. Além disso, a documentação é gerada dinamicamente, sendo atualizada a cada mudança no contrato da API.

As figuras 4 a 6 ilustram os contratos da API do metamodelo para aplicativos cívicos e sua visualização por meio da interface Swagger. As outras APIs – saúde, educação, assistência social e empregos – possuem interfaces de documentação semelhantes.

5. CONCLUSÃO E PERSPECTIVAS FUTURAS

Os aplicativos Mapa da Saúde, Nossa Escola, Meu Remédio, Mami e Vacin App constituem exemplos reais de aplicações desenvolvidas tendo por base a API disponibilizada. Estão disponíveis nas lojas *App Store* e *Google Play*.

Em 2016, o TCU promoveu mais uma oportunidade de validar o modelo para aplicativos cívicos aqui descrito: o **Desafio de Aplicativos Cívicos**⁸. Trata-se de uma *hackathon* para que desenvolvedores externos inscrevam aplicativos que façam uso dos *web services* disponibilizados.

Na época em que este artigo foi escrito, o concurso estava em andamento, com cerca de noventa aplicativos inscritos. Ao longo do Desafio, mantivemos um canal aberto diretamente com os desenvolvedores, através de caixa postal institucional e de um site no GitHub⁹. Por

meio desse canal, recebemos mais de quarenta sugestões de ajustes e extensões às operações disponibilizadas pela API. A maioria das sugestões foram implementadas, e algumas encontram-se na lista de melhorias a serem implementadas na próxima versão da API.

Embora tenha sido proposta e implementada pelo TCU, a plataforma de *web services* disponibilizada está aberta a contribuições e parcerias. Seu sucesso dependerá da convergência de esforços e ideias na construção de soluções inovadoras e úteis ao cidadão. Se bem-sucedida,

Figura 4:
Exemplo de listagem de *web services* (metamodelo para aplicativos cívicos)

AppCívicoRS
Web Services referentes ao modelo comum para aplicativos cívicos.

[Contact the developer](#)
[TCU](#)

aplicativos : Aplicativos	Show/Hide	List Operations	Expand Operations
grupos : Grupos	Show/Hide	List Operations	Expand Operations
hashtags : Hashtags	Show/Hide	List Operations	Expand Operations
instalacoes : Instalações	Show/Hide	List Operations	Expand Operations
notificacoes : Notificações	Show/Hide	List Operations	Expand Operations
peessoas : Pessoas	Show/Hide	List Operations	Expand Operations
postagens : Postagens	Show/Hide	List Operations	Expand Operations
tipos de objeto : Tipos de Objeto	Show/Hide	List Operations	Expand Operations
tipos de postagem : Tipos de Postagem	Show/Hide	List Operations	Expand Operations

Fonte: Elaboração própria

Figura 5:
Exemplo de listagem de operações de um *web service*

aplicativos : Aplicativos Show/Hide List Operations Expand Operations

GET	/rest/aplicativos	Retorna o conjunto de aplicativos cadastrados.
POST	/rest/aplicativos	Cadastra um novo aplicativo.
GET	/rest/aplicativos/pessoa/{codPessoa}	Retorna o conjunto de aplicativos de um responsável.
GET	/rest/aplicativos/{codAplicativo}	Recupera os dados de um determinado aplicativo pelo código.
PUT	/rest/aplicativos/{codAplicativo}	Atualiza os dados de um aplicativo já cadastrado.
GET	/rest/aplicativos/{codAplicativo}/hashtags	Recupera as hashtags cadastradas de um determinado aplicativo.
GET	/rest/aplicativos/{codAplicativo}/tipos-perfil	Recupera os tipos de perfil criados para um determinado aplicativo.
POST	/rest/aplicativos/{codAplicativo}/tipos-perfil	Cadastra um novo tipo de perfil para um dado aplicativo.
GET	/rest/aplicativos/{codAplicativo}/tipos-perfil/{codTipoPerfil}	Recupera os dados de um determinado tipo de perfil.
PUT	/rest/aplicativos/{codAplicativo}/tipos-perfil/{codTipoPerfil}	Atualiza os dados de um tipo de perfil já cadastrado.

Fonte: Elaboração própria

Figura 6:

Exemplo de detalhamento de contrato de uma operação

aplicativos : Aplicativos Show/Hide | List Operations | Expand Operations

GET /rest/aplicativos Retorna o conjunto de aplicativos cadastrados.

Response Class (Status 200)

Model | Model Schema

```

Aplicativo {
  cod (long, optional),
  descricao (string, optional),
  links (Array(Link), optional),
  nome (string, optional)
}

Link {
  href (string, optional),
  rel (string, optional),
  templated (boolean, optional)
}

```

Response Content Type application/json

Fonte: Elaboração própria

trará resultados que poderão contribuir para a melhoria dos serviços e políticas públicas, provendo novos serviços e recursos para a sociedade como um todo.

NOTAS

- 1 Disponível em: <<http://dados.gov.br>>. Acesso em: 10 out. 2016.
- 2 Disponível em: <<http://dados.gov.br/dataset/cnes>>. Acesso em: 10 out. 2016.
- 3 Disponível em: <<http://www.comprasgovernamentais.gov.br/aceso-aos-sistemas/comprasnet-siasg>>. Acesso em: 10 out. 2016.
- 4 Disponível em: <<http://www.tesouro.fazenda.gov.br/siafi>>. Acesso em: 10 out. 2016.
- 5 Disponível em: <<http://cne.smpe.gov.br/>>. Acesso em: 10 out. 2016.
- 6 Disponível em: <<http://www.bepiducb.com.br/index.html>>. Acesso em: 10 out. 2016.
- 7 Disponível em: <<http://swagger.io/>>. Acesso em: 10 out. 2016.
- 8 Disponível em: <<http://portal.tcu.gov.br/desafio-aplicativos-civicos/>>. Acesso em: 10 out. 2016.
- 9 Disponível em: <<https://github.com/>>. Acesso em: 10 out. 2016.

REFERÊNCIAS

- BOOTH, D. et al. (Eds.). W3C. Web Services Architecture. February 2004. Seção 1.4. Disponível em: <<https://www.w3.org/TR/ws-arch/#whatis>>. Acesso em: 10 out. 2016.
- BRASIL. Lei nº 12.527, de 18 de novembro de 2011. Dispõe sobre os procedimentos a serem observados pela União, Estados, Distrito Federal e Municípios, com o fim de garantir o acesso a informações, e dá outras providências. Diário Oficial da União, Brasília, DF, 18 nov. 2011. Edição extra.
- ECMA. The JSON Data Interchange Format. Geneva, out. 2013. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: 10 out. 2016.
- ERL, T. et al. SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST. New Jersey: Prentice Hall, 2012. p. 24.
- FIELDING, R. T. Architectural styles and the design of network-based software architectures. 2000. Dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Information and Computer Science. University of California, Irvine, 2000.
- FIELDING, R. et al. Hypertext Transfer Protocol – HTTP/1.1. Junho 1999. Disponível em: <<https://www.w3.org/Protocols/rfc2616/rfc2616.txt>>. Acesso em: 10 out. 2016.